**Project Description – Tech Guide**

**Tech Guide** is a web application designed to help students explore the world of software and technology in an organised, beginner-friendly way.  
It acts as a personal guide by:

* **Assessing interests and skills:** Students take a short online quiz to identify their strengths and preferences.
* **Recommending a tech path:** Based on quiz answers, the system suggests a suitable technology stack or domain (e.g., Frontend Development, Data Science, AI/ML).
* **Providing roadmaps:** Each recommended path comes with a clear step-by-step roadmap showing what to learn next.
* **Listing resources:** For every step in the roadmap, the app shows curated learning resources — both free and paid — so students can start immediately.

The platform will have a simple, responsive **frontend** (HTML/CSS/JavaScript), a **Django backend** to handle user data and quiz results, and a lightweight **AI component** to generate recommendations from quiz responses.

By combining an easy-to-use interface with personalised guidance, Tech Guide helps students make informed decisions about their learning journey in technology.  
  
**Tech Stack to be used :**

| **Layer** | **Technology** | **Why Use It** |
| --- | --- | --- |
| **Frontend** | **HTML, CSS, JavaScript** (plain or Bootstrap/Tailwind for styling) | Simple, beginner-friendly, quick to build. |
| **Backend Framework** | **Django (Python)** | Handles routing, user authentication, database, and quiz logic. |
| **Database** | SQLite (default with Django) or PostgreSQL (if you want production-ready) | Stores quiz questions, user info, results, roadmaps, resources. |
| **AI / Recommendation Engine** | Simple rule-based logic or a small scikit-learn model embedded in Django | Maps quiz answers to recommended domains. |
| **Version Control** | Git + GitHub | Team collaboration. |
| **Deployment (Optional)** | PythonAnywhere / Render / Heroku (free tiers) | Host your project online for demo. |

**Basic Flow of Development**

**Phase 1 – Frontend Mockup (Static)**

* Build **HTML/CSS/JS pages** for:
  + Home
  + Quiz
  + Result
  + Roadmaps/Resources
* Make sure nav links work (static pages).

**Phase 2 – Django Setup**

* Create a new Django project: django-admin startproject techguide
* Create app for quiz & recommendations: python manage.py startapp quiz
* Configure templates/static files to serve your HTML/CSS from Django.

**Phase 3 – Database + Models**

* Define models for:
  + Questions
  + Choices/Answers
  + Tech Domains (Frontend, Data Science…)
  + Roadmaps/Resources
* Use Django Admin to add data easily.

**Phase 4 – Quiz Logic**

* Create views to display quiz questions dynamically from DB.
* Handle quiz submission → collect answers.
* Implement **rule-based scoring or simple ML model** to decide recommended domain.
* Store results in DB linked to user.

**Phase 5 – Roadmaps & Resources**

* Create pages where, based on recommendation, you display:
  + Steps in roadmap
  + Curated resources from DB

**Phase 6 – User Accounts (Optional)**

* Add Django’s built-in authentication for registration/login so users can:
  + Save progress
  + Revisit results

**Phase 7 – AI Enhancement (Optional)**

* If time permits, replace rule-based logic with a small scikit-learn model:
  + Collect sample data: quiz answers → domain
  + Train model offline
  + Load model file in Django to make predictions.

**Phase 8 – Styling + Testing**

* Make UI responsive (Bootstrap/Tailwind).
* Test full flow: register → take quiz → see recommendation → roadmap.

**Phase 9 – Deployment (Optional)**

* Push code to GitHub.
* Deploy on PythonAnywhere/Render/Heroku for demo.